Балаковский инженерно-технологический институт - филиал

федерального государственного автономного образовательного учреждения

высшего образования

«Национальный исследовательский ядерный университет «МИФИ»

Факультет атомной энергетики и технологий

Кафедра «Информатика и управление в технических системах»

ПРАКТИЧЕСКАЯ РАБОТА №1

по дисциплине

«Языки программирования»

Вариант №9

Выполнил: студент группы ИФСТ-2з

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Никольский А.Г.

« 26 » ноября 2020г.

|  |  |
| --- | --- |
|  | Проверил ст. преподаватель каф. ИУС  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Михеев И.В.  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 202\_\_\_г. |

Балаково 2020

Цель работы: изучение основ языка Си на примере реализации линейных алгоритмов. Получение базовых навыков работы в интегрированной среде разработки Microsoft Visual Studio 2005.

Задание №1.

Окно выполнения программы изображено на рис. 1.

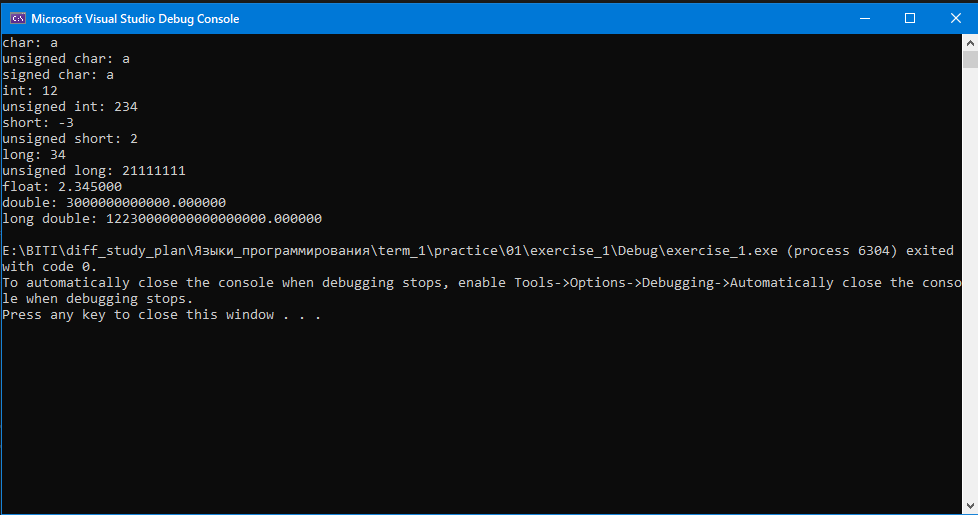


Рисунок 1 - Окно выполнение программы задания №1

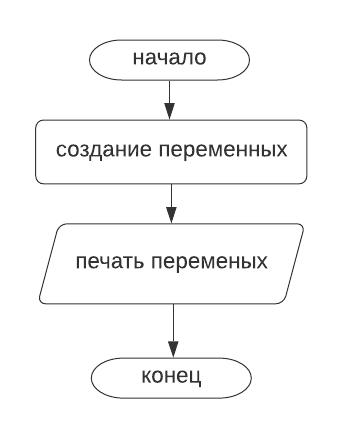
Блок схема алгоритма изображена на рис.2.

Рисунок 2 - Блок схема к заданию №1

Исходный текст программы:

// Created by a.nikolsky 26.11.2020

#include <iostream>

int main()

{

//Basic types:

//integer types

char c = 'a';

unsigned char uc = 'a';

signed char sc = 'a';

int i = 12;

unsigned int ui = 234;

short s = -3;

unsigned short us = 2;

long l = 34;

unsigned long ul = 21111111;

//floating-point types

float f = 2.345;

double d = 3e12;

long double ld = 12.23e18;

//the void type

//The void type specifies that no value is available.

//It is used in three kinds of situations.

//1. Function returns as void: void exit (int status);

//2. Function arguments as void: int rand(void);

//3. Pointers to void: void \*malloc( size\_t size )

printf("char: %c \n", c);

printf("unsigned char: %c \n", uc);

printf("signed char: %c \n", sc);

printf("int: %d \n", i);

printf("unsigned int: %u \n", ui);

printf("short: %hi \n", s);

printf("unsigned short: %hu \n", us);

printf("long: %li \n", l);

printf("unsigned long: %lu \n", ul);

printf("float: %f \n", f);

printf("double: %f \n", d);

printf("long double: %f \n", ld);

// ------------------------------------------------------

//Derived types:

// Array types

int arr[5] = { 1,2,3,4,5 };

// Struct types

struct s {

int i;

double d;

const char\* c;

} s1{ 1, 1.1,"qqqqq" };

// Pointer types

int\* p\_int = &i;

//Union types

union Data {

int i;

float f;

char str[20];

} data;

//3. Pointers to void: void \*malloc( size\_t size )

// ------------------------------------------------------

// Enumeration types

enum State { Working = 1, Failed = 0 } state = Working;

}

Задание №2.

Окно выполнения программы изображено на рис. 3.

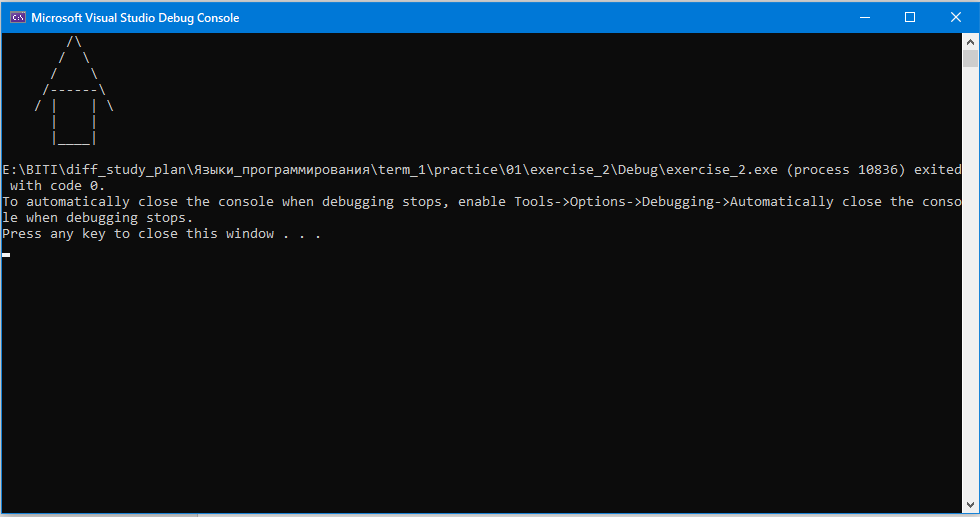


Рисунок 3 - Окно выполнение программы задания №2

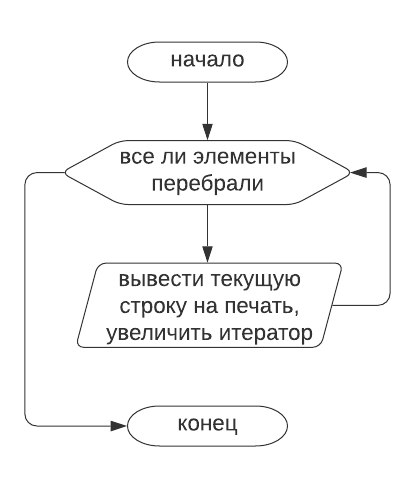
Блок схема алгоритма изображена на рис.4.

Рисунок 4 - Блок схема к заданию №2

Исходный код программы:

// created by a.nikiolsky 27.11.2020

#include <iostream>

#include <vector>

const std::vector<std::string> IMAGE =

{" /\\ \n",

" / \\ \n",

" / \\ \n",

" /------\\ \n",

" / | | \\ \n",

" | | \n",

" |\_\_\_\_| \n",

};

int main()

{

for (auto& string : IMAGE) {

std::cout << string;

}

}

Задание №3.

Окно выполнения программы изображено на рис.5.

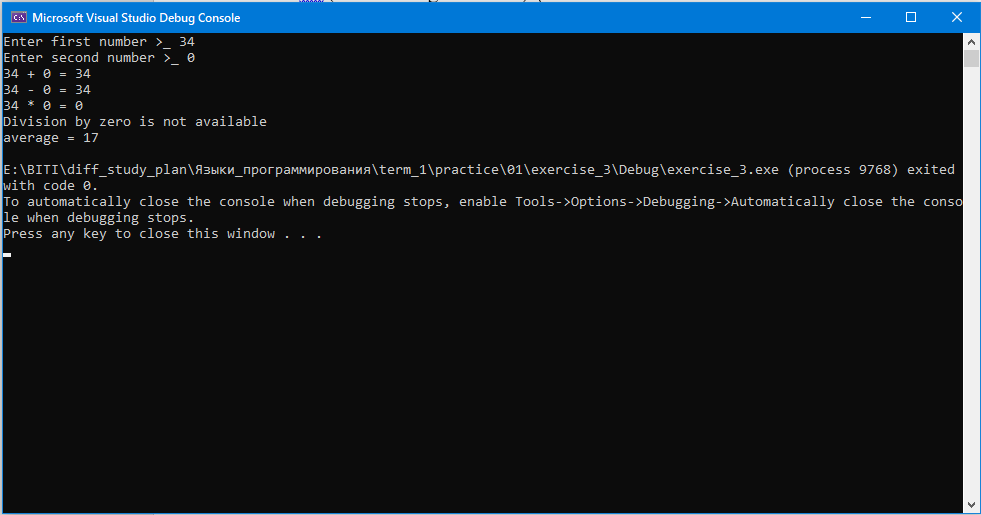


Рисунок 5 - Окно выполнение программы задания №3

Блок схема алгоритма изображена на рис.6

Рисунок 6 - Блок схема к заданию №3

Исходный код программы:

// created by a.nikolsky 27.11.2020

#include <iostream>

#include <string>

#include <sstream>

template <typename T>

T get\_value(std::string&& message, bool(\*compare)(const T&) = compare\_default) {

T result;

std::string buff = "";

std::stringstream ss;

do {

ss.clear();

buff = "";

std::cout << message;

std::getline(std::cin, buff);

ss.str(buff);

ss >> result;

} while (ss.fail() || !compare(result));

return result;

};

template <typename T>

bool compare\_default(const T& value = 0) {

return true;

}

void printResult(const int& a, const int& b) {

std::cout << a << " + " << b << " = " << a + b << std::endl;

std::cout << a << " - " << b << " = " << a - b << std::endl;

std::cout << a << " \* " << b << " = " << a \* b << std::endl;

if (b != 0) {

std::cout << a << " / " << b << " = " << a / b << std::endl;

}

else {

std::cout << "Division by zero is not available" << std::endl;

}

std::cout << "average = " << (a + b) / 2 << std::endl;

}

int main()

{

int first = get\_value<int>("Enter first number >\_ ");

int second = get\_value<int>("Enter second number >\_ ");

printResult(first, second);

}

Задание №4.

Окно выполнения программы изображено на рис. 7.

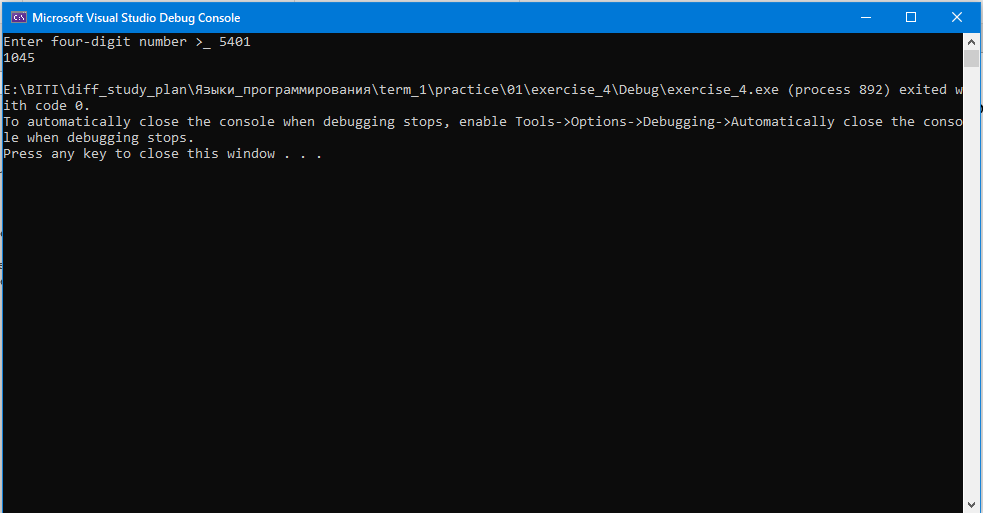


Рисунок 7 - Окно выполнение программы задания №4

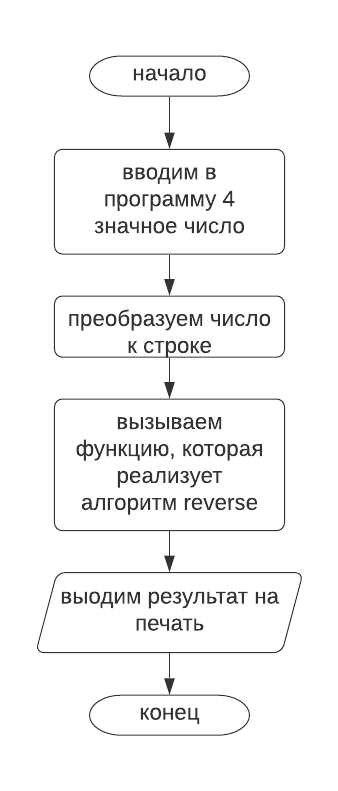
Блок схема алгоритма изображена на рис.8.

Рисунок 8 - Блок схема к заданию №4

Исходный код программы:

// created by a.nikolsky 27.11.2020

#include <iostream>

#include <string>

#include <sstream>

#include <algorithm>

template <typename T>

T get\_value(std::string&& message, bool(\*compare)(const T&) = compare\_default) {

T result;

std::string buff = "";

std::stringstream ss;

do {

ss.clear();

buff = "";

std::cout << message;

std::getline(std::cin, buff);

ss.str(buff);

ss >> result;

} while (ss.fail() || !compare(result));

return result;

};

template <typename T>

bool compare\_default(const T& value = 0) {

return true;

}

bool compare\_number\_count(const int& number) {

if (number < 1000 || number > 9999) {

return false;

}

else {

return true;

}

}

int main()

{

int number = get\_value<int>("Enter four-digit number >\_ ", compare\_number\_count);

std::string str = std::to\_string(number);

std::reverse(str.begin(), str.end());

std::cout << str << std::endl;

}

Задание №5

Окно выполнения программы изображено на рис. 9.

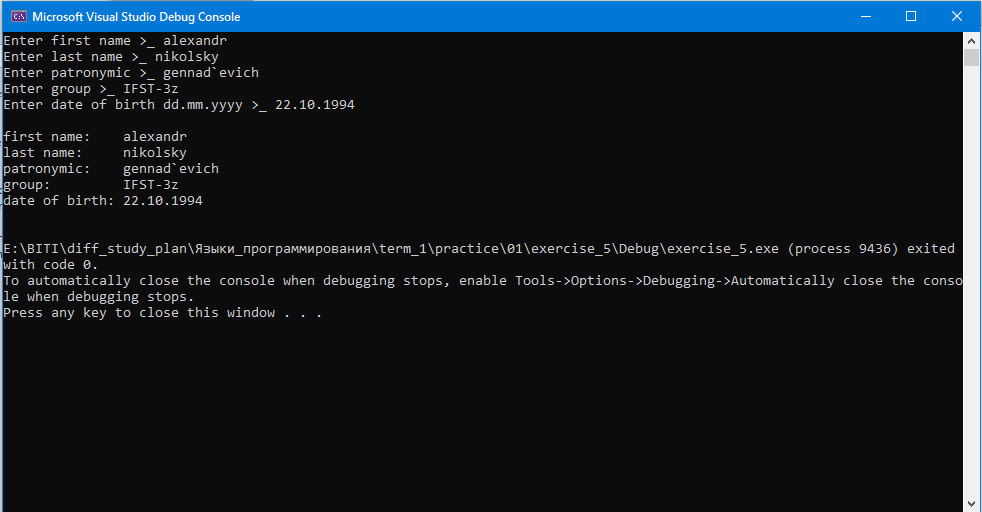


Рисунок 9 - Окно выполнение программы задания №5

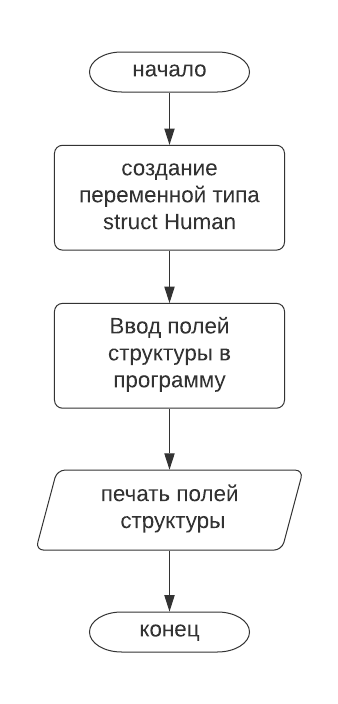
Блок схема алгоритма изображена на рис.10.

Рисунок 10 - Блок схема к заданию №5

Исходный код программы:

// created by a.nikolsky 27.11.2020

#include <iostream>

#include <string>

struct Human{

std::string first\_name;

std::string last\_name;

std::string patronymic;

std::string group;

std::string date\_of\_birth;

};

std::ostream& operator << (std::ostream& stream, const Human& h) {

stream << std::endl;

stream << "first name: " << h.first\_name << std::endl;

stream << "last name: " << h.last\_name << std::endl;

stream << "patronymic: " << h.patronymic << std::endl;

stream << "group: " << h.group << std::endl;

stream << "date of birth: " << h.date\_of\_birth << std::endl;

return stream;

}

void get\_human\_information(Human& h) {

std::cout << "Enter first name >\_ ";

std::getline(std::cin, h.first\_name);

std::cout << "Enter last name >\_ ";

std::getline(std::cin, h.last\_name);

std::cout << "Enter patronymic >\_ ";

std::getline(std::cin, h.patronymic);

std::cout << "Enter group >\_ ";

std::getline(std::cin, h.group);

std::cout << "Enter date of birth dd.mm.yyyy >\_ ";

std::getline(std::cin, h.date\_of\_birth);

return;

}

int main()

{

Human h;

get\_human\_information(h);

std::cout << h << std::endl;

}

Задание №6.

Окно выполнения программы изображено на рис. 11.

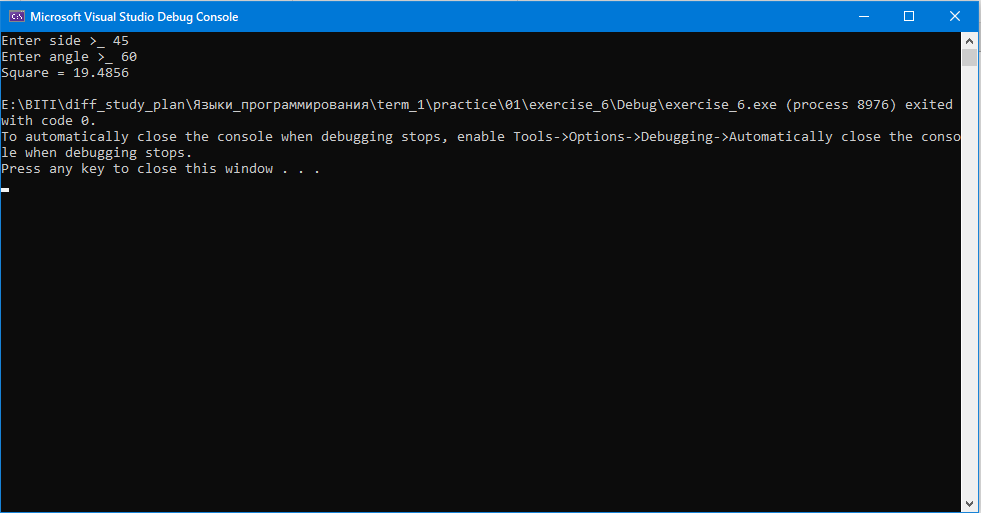


Рисунок 11 - Окно выполнение программы задания № 6

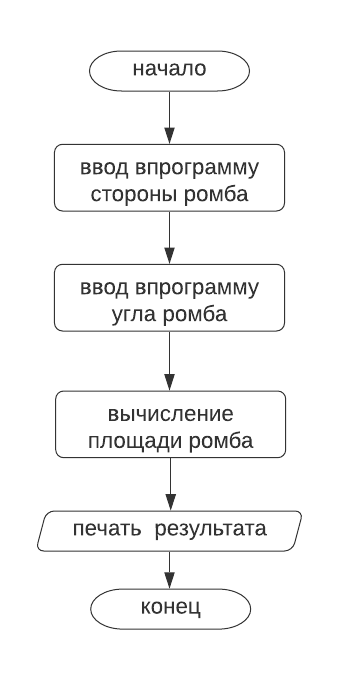
Блок схема алгоритма изображена на рис.12.

Рисунок 12 - Блок схема к заданию №6

Исходный код программы:

// created by a.nikolsky 27.11.2020

#define \_USE\_MATH\_DEFINES

#include <iostream>

#include <math.h>

#include <string>

#include <sstream>

double rhomb\_square(const double& side, const double& angle) {

return 0.5 \* side \* sin(angle \* M\_PI / 180);

}

template <typename T>

T get\_value(std::string&& message, bool(\*compare)(const T&) = compare\_default) {

T result;

std::string buff = "";

std::stringstream ss;

do {

ss.clear();

buff = "";

std::cout << message;

std::getline(std::cin, buff);

ss.str(buff);

ss >> result;

} while (ss.fail() || !compare(result));

return result;

};

template <typename T>

bool compare\_default(const T& value = 0) {

return true;

}

bool compare\_angle(const double& angle) {

if (angle >= 180 || angle <= 0) {

return false;

}

else{

return true;

}

}

bool compare\_side(const double& side) {

if (side <= 0) {

return false;

}

else {

return true;

}

}

int main()

{

double side = get\_value<double>("Enter side >\_ ", compare\_side);

double angle = get\_value<double>("Enter angle >\_ ", compare\_angle);

std::cout << "Square = " << rhomb\_square(side, angle) << std::endl;

}

Вывод: в данной работе изучили основы языка Си (С++) на примере реализации линейных алгоритмов. Получили базовые навыки работы в интегрированной среде разработки Microsoft Visual Studio 2005.